Scientia Iranica A (2011) 18 (4), 835–845

Sharif University of Technology

**Scientia Iranica**

*Transactions A: Civil Engineering*

www.sciencedirect.com

# Discrete Least Squares Meshless (DLSM) method for simulation of steady state shallow water flows

**A. Rahmani Firoozjaee** [a,*], **M.H. Afshar** [b]

[a] *Faculty of Civil Engineering, Iran University of Science and Technology, Tehran, Iran*

[b] *Enviro-Hydroinformatic Center of Excellence, Faculty of Civil Engineering, Iran University of Science and Technology, Tehran, Iran*

Received 17 January 2010; revised 14 May 2011; accepted 16 July 2011

**KEYWORDS**

Discrete Least Squares Meshless (DLSM);
Shallow water;
Steady state flows.

**Abstract** In this paper, a meshless method, namely, Discrete Least Squares Meshless (DLSM) method is used for the solution of shallow water problems. In this method, the computational domain is discretized by some nodes and then a set of simultaneous equations are built using Moving Least Squares (MLS) shape functions and the least squares technique. The proposed method does not need any background mesh and therefore is a truly meshless method. The stability and accuracy of the DLSM method is improved using some sampling points. Some 2D benchmark problems and a two dimensional flow over an ogee spillway are used to illustrate the performance of the present DLSM method. Numerical results are compared with experimental or analytical ones. Both regular and irregular meshes of nodes are used to show the potential of the DLSM method in solving problems with more complex domains.



## 1. Introduction

Shallow water equations, derived from three-dimensional incompressible Navier–Stokes equations using the hydrostatic pressure assumption, are a standard mathematical representation, valid for most types of flow encountered in coastal sea, river, channel and ocean modeling. They can be utilized to predict storm surges, tsunamis and floods. Shallow water equations may be combined with some additional equations (e.g., transport, reaction) for the simulation of contaminant propagation, temperature and salinity transport and other engineering problems.

Several numerical solutions have been proposed for nonlinear shallow water equations. Cooley in 1976 used the standard Galerkin finite element method to solve 1D shallow water, Saint Venant, equations [1]. Some researchers solved shallow water equations using the finite difference method [2,3]. Open channel flow was analyzed using shallow water equations and applying the finite volume method [4]. A two dimensional numerical model of spillway flow was proposed by Unami et al. [5]. The discontinuous Galerkin finite element method was applied for the solution of shallow water equations in [6]. Zhou and coworkers proposed a meshless method to solve shallow water equations using RBF shape functions [11]. Shallow water equations were analyzed using the discrete least squares method with structured nodes [12]. The Natural Element Method (NEM), in a fully Lagrangian formulation, was used to simulate shallow water flow in the presence of strong gradients [9]. The shallow water equations were formulated by a variable smoothing length SPH method for the solution of dam break problems on different geometries [10].

Over the last decade, meshless methods for the solution of Partial Differential Equations (PDEs) have become increasingly popular. The main idea of these methods is to approximate the unknown field by a linear combination of shape functions, built without having recourse to a mesh of the domain. Instead, nodes are scattered in the domain, and a certain weight function with a local support is associated with each of these nodes. The shape function associated with a given node is then built considering weight functions whose support overlaps the weight function of this node.

* Corresponding address: Department of Civil Engineering, Babol Noushirvan University of Technology, Babol, P.O. Box 484, Mazandaran, Iran.
*E-mail address:* ARahmani@IUST.ac.ir (A. Rahmani Firoozjaee).

1026-3098 © 2011 Sharif University of Technology. Production and hosting by Elsevier B.V. Open access under CC BY license.
Peer review under responsibility of Sharif University of Technology.

doi:10.1016/j.scient.2011.07.016

Production and hosting by Elsevier

Several meshless methods have been proposed in the literature, many of them in the last two decades. Meshless methods that have been developed in recent years are Smoothed Particle Hydrodynamics (SPH) [13], Element Free Galerkin (EFG) [14,15], Reproducing Kernel Particle (RKP) [16], Finite Point (FP) [17], hp-clouds [18], Meshless Local Petrov–Galerkin (MLPG) [19–23], Local Boundary Integral Equation (LBIE) [24,7] and finite cloud [8].

Some meshless methods, named collocation methods, are methods that directly use the governing equation. These methods are often arrived at using a point collocation weighted residual formulation of the problem. The collocation methods are simpler in application; however, they suffer from stability problems. A Meshless Galerkin Least Squares (MGLS) method, proposed by Pan et al. [25], combining the advantages of the Galerkin and collocation methods. In this approach, the Galerkin method is applied on the boundary, whereas the least squares technique is applied to the interior domain. To alleviate the instability difficulties of collocation methods, a Least Squares Collocation Meshless (LSCM) method [26] is developed. Except for the nodes used to construct the shape functions, a number of auxiliary points were also used. The system of discretized equations was constructed by collocating the differential equation and the boundary conditions at these sampling points. The number of resulting equations was consequently greater than that of unknowns. Therefore, the set of equations was solved using the least squares method. Numerical studies showed that LSCM is stable and efficient with high accuracy, but its coefficient matrix is asymmetric.

The Discrete Least Squares Meshless (DLSM) method is proposed by Arzani and Afshar [27,28]. It shows more stability than the collocation method, but on unstructured nodes, the results are no of high quality. To improve the stability and accuracy of the method, it has been modified by Firoozjaee and Afshar [29], using some sampling points. This method is based on minimizing the least squares functional defined by the sum of squared residuals of the differential equation and its boundary condition at some points, called sampling points, over the domain and its boundary. The sampling points are generally considered to be different from the nodal points used to discretize the problem domain. In this method, the number of equations is identical to the number of field nodes. It provides a symmetric coefficient matrix and preserves high accuracy and stability, even for irregular nodes. Full details of the DLSM method with sampling points can be found in [29] for elliptic partial differential and in [30] for 1D hyperbolic problems. The method does not need any background mesh and therefore can be considered a truly meshless method.

In this paper, the discrete least squares meshless method is applied for the solution of shallow water equations using an unstructured mesh of nodes. This will be very useful for modeling real engineering problems. In what follows, Section 2 states a brief description of the DLSM method. Application of the DLSM method to shallow water equations is described in Section 3. In Section 4, some benchmark numerical examples are solved using different meshes of regular and irregular nodes. And finally Section 5 deals with the solution of flow over an ogee spillway.

## 2. Discrete least squares meshless (DLSM) method

A brief description of the Discrete Least Squares Meshless (DLSM) method is presented, considering a general form of differential equations and its appropriate boundary conditions, as follows [29]:

$$\Im(\varphi) = f \quad \text{in } \Omega, \tag{1}$$

$$\phi - \bar{\phi} = 0 \quad \text{on } \Gamma_1, \tag{2}$$

$$\Re(\phi) - \bar{t} = 0 \quad \text{on } \Gamma_2 \tag{3}$$

in which $\Im$ and $\Re$ are differential operators that describe the governing equation and the Neumann boundary condition on $\Gamma_2$, respectively. $\Gamma_1$ represents the Dirichlet boundary condition with a prescribed value of $\bar{\phi}$, and $f$ denotes the source term.

Function approximation is carried out using the standard Moving Least Squares (MLS) shape functions initially presented by Shepard [31]. The least squares functional of the partial differential equation and its boundary condition is stated as:

$$\begin{aligned} I = & \sum_{k=1}^{ms} \left[ \sum_{j=1}^{m} \Im(N_j(X_k))\varphi_j - f(K_k) \right]^2 \\ & + \alpha_1 \sum_{k=1}^{md} \left[ \sum_{j=1}^{m} N_j(X_k)\varphi_j - \bar{\varphi}(K_k) \right]^2 \\ & + \alpha_2 \sum_{k=1}^{mn} \left[ \sum_{j=1}^{m} \Re(N_j(X_k))\varphi_j - \bar{t}(K_k) \right]^2, \end{aligned} \tag{4}$$

where $m$ is the total number of nodes used to discretize the problem domain and its boundaries, here referred to as field nodes. $N_j$ and $\phi_j$ represent the shape function and nodal parameter of node $j$, respectively. $ms$ denotes the number of sampling points in the computational domain, and $md$ and $mn$ are the number of sampling points on the Dirichlet and Neumann boundaries, respectively. $\alpha_1$ and $\alpha_2$ are penalty coefficients for Dirichlet and Neumann boundary conditions, respectively. Minimization of the least squares functional, Eq. (4), with respect to the nodal parameters ($\phi_j$, $j = 1, 2, \ldots, m$) leads to the following Symmetric and Positive Definite (SPD) system of equations:

$$\mathbf{K\Phi} = \mathbf{F}, \tag{5}$$

in which $\mathbf{\Phi}$ is the vector of the nodal parameter and:

$$\begin{aligned} K_{ij} = & \sum_{k=1}^{ms} \Im(N_i^T(X_k))\Im(N_j(X_k)) + \alpha_1 \sum_{k=1}^{md} N_i^T(X_k)N_j(X_k) \\ & + \alpha_2 \sum_{k=1}^{mn} \Re(N_i^T(X_k))\Re(N_j(X_k)), \end{aligned} \tag{6}$$

$$\begin{aligned} F_i = & \sum_{k=1}^{ms} \Im(N_i^T(X_k))f(X_k) + \alpha_1 \sum_{k=1}^{md} N_i^T(X_k)\bar{\varphi}(X_k) \\ & + \alpha_2 \sum_{k=1}^{mn} \Re(N_i^T(X_k))\bar{t}(X_k). \end{aligned} \tag{7}$$

It can be seen clearly that the obtained stiffness matrix is symmetric, even for none self-adjoint differential operators. The symmetric positive definite properties of the coefficient matrix allow the use of efficient solvers for the solution of the resulting set of algebraic equations.

## 3. Shallow water equations

The flow of water in shallow layers, such as what occurs in coastal estuaries, oceans, rivers, channels etc., is an important discipline in water engineering. The shallow water model is based on assumptions of incompressible flow with a hydrostatic pressure distribution and a uniform velocity distribution in depth. These equations, sometimes named depth average equations, are obtained by integrating Navier–Stokes equations over the water depth. Eq. (8) presents two dimensional shallow water equations [32]:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial (hu)}{\partial x} + \dfrac{\partial (hv)}{\partial y} = 0 \\ \dfrac{\partial (hu)}{\partial t} + \dfrac{\partial (hu^2 + gh^2/2)}{\partial x} + \dfrac{\partial (huv)}{\partial y} - gh(s_{0x} - s_{fx}) = 0 \\ \dfrac{\partial (hv)}{\partial t} + \dfrac{\partial (huv)}{\partial x} + \dfrac{\partial (hv^2 + gh^2/2)}{\partial y} - gh(s_{0y} - s_{fy}) = 0 \end{cases} \quad (8)$$

where $h$ is the flow depth, $u$ and $v$ are the depth average velocities in $x$ and $y$ directions, respectively, $g$ is the gravitational acceleration, $s_0$ and $s_f$ are bed and friction (energy line) slopes, and subscripts $x$ and $y$ denote the $x$ and $y$ component of slopes. In practice, usually, energy slopes would be written in terms of the Manning roughness coefficient:

$$s_{fx} = \frac{n^2 u\sqrt{u^2 + v^2}}{h^{4/3}}, \quad (9)$$

$$s_{fy} = \frac{n^2 v\sqrt{u^2 + v^2}}{h^{4/3}}, \quad (10)$$

where $n$ is the Manning's roughness coefficient. Neglecting energy slopes and representing shallow water equations in matrix notation yields the compact form:

$$\mathbf{U}_{,t} + \mathbf{A}\mathbf{U}_{,x} + \mathbf{B}\mathbf{U}_{,y} = \mathbf{S}\mathbf{U}, \quad (11)$$

in which $U = \begin{pmatrix} h & hu & hv \end{pmatrix}^T$ is the vector of unknowns, and $A$, $B$ and $S$ are matrices as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad (12)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + gh & 0 & 2v \end{bmatrix}, \quad (13)$$

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 0 \\ gs_{0x} & 0 & 0 \\ gs_{0y} & 0 & 0 \end{bmatrix}. \quad (14)$$

In the steady state case, shallow water equations may be reduced to Eq. (15):

$$\mathbf{A}\mathbf{U}_{,x} + \mathbf{B}\mathbf{U}_{,y} = \mathbf{S}\mathbf{U}. \quad (15)$$

It is notable that the shallow water equations are elliptic partial differential equations when the flow is subcritical and hyperbolic when the flow is supercritical. Imposition of the boundary condition is essentially related to characteristic lines [33].

Each steady state problem may be solved using two different algorithms named steady state and time marching. In the following parts, these algorithms for the DLSM method are described for the solution of steady state shallow water equations.

### 3.1. Steady state algorithm

Steady state shallow water equations (Eq. (15)) may be stated as:

$$-\mathbf{S}\mathbf{U} + \mathbf{A}\mathbf{U}_{,x} + \mathbf{B}\mathbf{U}_{,y} = 0. \quad (16)$$

Its boundary condition is the Dirichlet type boundary as follows:

$$\mathbf{U} - \bar{\mathbf{U}} = \mathbf{0} \quad \text{on } \Gamma_1. \quad (17)$$

The residual of the partial differential at typical point $k$, using moving least squares shape functions, is:

$$\mathbf{R}(\mathbf{x}_k) = -\mathbf{S}\sum_{j=1}^{m} \mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j + \mathbf{A}\sum_{j=1}^{m} \mathbf{N}_{j,x}(\mathbf{x}_k)\mathbf{U}_j + \mathbf{B}\sum_{j=1}^{m} \mathbf{N}_{j,y}(\mathbf{x}_k)\mathbf{U}_j^1, \quad (18)$$

where $\mathbf{N}_j(\mathbf{x}_k)$ is the MLS shape function of node $j$ at point $k$. $\mathbf{N}_{j,x}$ and $\mathbf{N}_{j,y}$ are derivatives of the shape function of node $j$, with respect to $x$ and $y$ axes, respectively, and $m$ is the total number of field nodes. As MLS shape functions do not satisfy the Kronecker delta property, the Dirichlet boundary condition can be applied using a penalty formulation. The residual of Eq. (17) is presented in Eq. (19):

$$\mathbf{R}_1(\mathbf{x}_k) = \sum_{j=1}^{m} \mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j - \bar{\mathbf{U}}. \quad (19)$$

A penalty approach is used to form the total residual of the problem defined as:

$$I = \sum_{k=1}^{ms} \mathbf{R}^2(\mathbf{x}_k) + \alpha_1 \sum_{k=1}^{md} \mathbf{R}_1^2(\mathbf{x}_k). \quad (20)$$

Using Eqs. (18) and (19), the above equation can be written as:

$$I = \sum_{k=1}^{ms} \left[ \sum_{j=1}^{m} -\mathbf{S}\mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j + \sum_{j=1}^{m} \mathbf{A}\mathbf{N}_{j,x}(\mathbf{x}_k)\mathbf{U}_j + \sum_{j=1}^{m} \mathbf{B}\mathbf{N}_{j,y}(\mathbf{x}_k)\mathbf{U}_j \right]^2 + \alpha_1 \sum_{k=1}^{md} \left[ \sum_{j=1}^{m} \mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j - \bar{\mathbf{U}} \right]^2, \quad (21)$$

where $ms$ is the number of sampling points, $md$ is the number of sampling points on the Dirichlet boundary, and $\alpha_1$ is the penalty coefficient for the Dirichlet boundary condition. Minimization of the functional, with respect to nodal parameters ($\mathbf{U}_j$, $j = 1, 2, \ldots, m$) leads to the following system of equations:

$$\mathbf{K}\mathbf{U} = \mathbf{F}, \quad (22)$$

where:

$$\mathbf{\Lambda}_i(\mathbf{x}_k) = -\mathbf{S}\mathbf{N}_i(\mathbf{x}_k) + \mathbf{A}\mathbf{N}_{i,x}(\mathbf{x}_k) + \mathbf{B}\mathbf{N}_{i,y}(\mathbf{x}_k), \quad (23)$$

$$\mathbf{K}_{ij} = \sum_{k=1}^{m} \mathbf{\Lambda}_i(\mathbf{x}_k)\mathbf{\Lambda}_j(\mathbf{x}_k) + \alpha_1 \sum_{k=1}^{md} \mathbf{N}_i(\mathbf{x}_k)\mathbf{N}_j(\mathbf{x}_k), \quad (24)$$

$$\mathbf{F}_i = \sum_{k=1}^{m} \mathbf{\Lambda}_i(\mathbf{x}_k)\mathbf{U}(\mathbf{x}_k) + \alpha_1 \sum_{k=1}^{md} \mathbf{N}_i(\mathbf{x}_k)\bar{\mathbf{U}}(\mathbf{x}_k). \quad (25)$$

### 3.2. Time marching algorithm

To solve steady state shallow water equations, one can solve the transient shallow water equation to reach a steady solution. Therefore, in this algorithm, Eq. (11) is temporarily discretized using an implicit method as follows:

$$\frac{\mathbf{U}^{n+1}-\mathbf{U}^{n}}{\Delta t}+\mathbf{A}\mathbf{U}_{,x}^{n+1}+\mathbf{B}\mathbf{U}_{,y}^{n+1}=\mathbf{S}\mathbf{U}^{n+1}, \tag{26}$$

where $\Delta t = t^{n+1} - t^{n}$ is the time increment.

After rearranging Eq. (26), the shallow water equation and its boundary condition can be written in the form of Eqs. (27) and (28):

$$(\mathbf{I}-\mathbf{S}\Delta t)\mathbf{U}^{n+1}+\mathbf{A}\Delta t\mathbf{U}_{,x}^{n+1}+\mathbf{B}\Delta t\mathbf{U}_{,y}^{n+1}=\mathbf{U}^{n} \quad \text{in } \Omega, \tag{27}$$

$$\mathbf{U}^{n+1}-\bar{\mathbf{U}}=\mathbf{0} \quad \text{on } \Gamma_1, \tag{28}$$

where $\mathbf{I}$ denotes the unit matrix and Eq. (28) represents the Dirichlet boundary condition. The residual of the partial differential equation at typical point $k$, using moving least squares shape functions, is:

$$\mathbf{R}(\mathbf{x}_k) = (\mathbf{I}-\mathbf{S}\Delta t)\sum_{j=1}^{m}\mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j^{n+1}+\mathbf{A}\Delta t\sum_{j=1}^{m}\mathbf{N}_{j,x}(\mathbf{x}_k)\mathbf{U}_j^{n+1} + \mathbf{B}\Delta t\sum_{j=1}^{m}\mathbf{N}_{j,y}(\mathbf{x}_k)\mathbf{U}_j^{n+1}-\mathbf{U}^{n}, \tag{29}$$

where $N_j(x_k)$ is the MLS shape function of node $j$ at point $k$. $N_{j,x}$ and $N_{j,y}$ are derivatives of the shape function of node $j$, with respect to $x$ and $y$ axes, respectively, and $m$ is the total number of field nodes. The residual of Eq. (28) is presented in Eq. (30);

$$\mathbf{R}_1(\mathbf{x}_k)=\sum_{j=1}^{m}\mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j^{n+1}-\bar{\mathbf{U}}, \tag{30}$$

where a penalty approach is used to form the total residual of the problem, defined as:

$$I=\sum_{k=1}^{ms}\mathbf{R}^2(\mathbf{x}_k)+\alpha_1\sum_{k=1}^{md}\mathbf{R}_1^2(\mathbf{x}_k). \tag{31}$$

Using Eqs. (29) and (30), the above equation can be written as Eq. (32):

$$I = \sum_{k=1}^{ms}\left[\sum_{j=1}^{m}(\mathbf{I}-\mathbf{S}\Delta t)\mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j^{n+1}+\sum_{j=1}^{m}\mathbf{A}\Delta t\mathbf{N}_{j,x}(\mathbf{x}_k)\mathbf{U}_j^{n+1} + \sum_{j=1}^{m}\mathbf{B}\Delta t\mathbf{N}_{j,y}(\mathbf{x}_k)\mathbf{U}_j^{n+1}-\mathbf{U}^{n}\right]^2 + \alpha_1\sum_{k=1}^{md}\left[\sum_{j=1}^{m}\mathbf{N}_j(\mathbf{x}_k)\mathbf{U}_j^{n+1}-\bar{\mathbf{U}}\right]^2, \tag{32}$$

where $ms$ is the number of sampling points, $md$ is the number of collocation (sampling) points on the Dirichlet boundary, and $\alpha_1$ is the penalty coefficient for the Dirichlet boundary condition. Minimization of the functional, with respect to nodal parameters ($U_j^{n+1}$, $j = 1, 2, \ldots, m$), leads to the following system of equations:

$$\mathbf{K}\mathbf{U}^{n+1}=\mathbf{F}, \tag{33}$$

where:

$$\mathbf{\Lambda}_i(\mathbf{x}_k)=(\mathbf{I}-\mathbf{S}\Delta t)\mathbf{N}_i(\mathbf{x}_k)+\mathbf{A}\Delta t\mathbf{N}_{i,x}(\mathbf{x}_k)+\mathbf{B}\Delta t\mathbf{N}_{i,y}(\mathbf{x}_k), \tag{34}$$

$$\mathbf{K}_{ij}=\sum_{k=1}^{m}\mathbf{\Lambda}_i(\mathbf{x}_k)\mathbf{\Lambda}_j(\mathbf{x}_k)+\alpha_1\sum_{k=1}^{md}\mathbf{N}_i(\mathbf{x}_k)\mathbf{N}_j(\mathbf{x}_k), \tag{35}$$

$$\mathbf{F}_i=\sum_{k=1}^{m}\mathbf{\Lambda}_i(\mathbf{x}_k)\mathbf{U}^{n}(\mathbf{x}_k)+\alpha_1\sum_{k=1}^{md}\mathbf{N}_i(\mathbf{x}_k)\bar{\mathbf{U}}(\mathbf{x}_k). \tag{36}$$

The system of equations stated by Eqs. (22) and (33) clearly allows efficient solvers to be used. It is notable that increasing $m$ for obtaining better results, yields to an increase in the dimension of the stiffness matrix, but increasing the number of collocation (sampling) points yields better results without increasing the dimension of the stiffness matrix. Therefore, it has little effect on computational cost.

The implicit time marching algorithm presented here can be implemented in two different ways [32]. In the first method, the coefficients of the semi-discretized shallow water equations (Eq. (27)) are considered at time $t^{n+1}$, leading to a nonlinear system of algebraic equations to be solved iteratively. This method is used when a transient solution using large time step sizes is required. In the second approach, used here, coefficients of Eq. (27) are considered at time $t^{n}$, leading to a linear system of equations upon spatial discretization. This method is not suitable for a transient solution and can only be used to arrive at steady state solutions. It should be noted that both implementations use an implicit temporal discretization and therefore are unconditionally stable. No time step size variations are required in this implementation, and a constant time step size can be, and has been, used.

## 4. Numerical examples

Shallow water equations have a hyperbolic nature. Thus, in this section, several two dimensional hyperbolic problems are solved using the DLSM method to represent the ability of the method, in confront with regular and irregular nodes in hyperbolic problems.

### 4.1. Linear advection

A two dimensional, linear advection equation, with a scalar variable, is given by Eq. (37):

$$\phi_{,x}+\phi_{,y}=0. \quad \text{in } 0\le x\le 1,\ 0\le y\le 1. \tag{37}$$

With the following boundary condition:

$$\phi=0 \quad \text{on } x=0,\ 0\le y\le 1, \tag{38}$$

$$\phi=1 \quad \text{on } y=0,\ 0\le x\le 1. \tag{39}$$

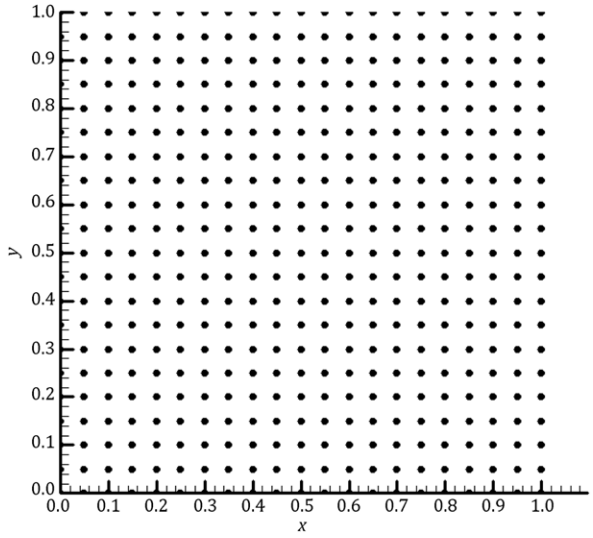


Figure 1: Evenly distributed field nodes in a [0 1]×[0 1] computational domain.

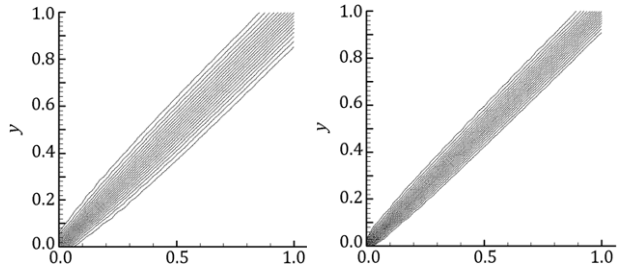


Figure 2: Solution of linear advection.

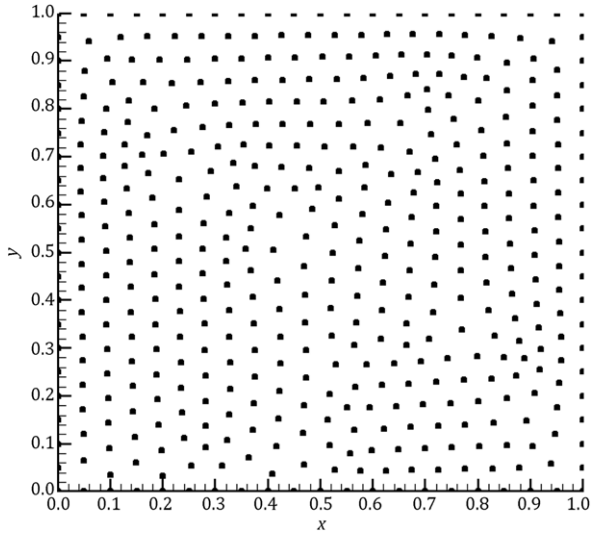


Figure 3: Irregularly distributed field nodes in a [0 1] × [0 1] computational domain.

This partial differential equation is identical to Eq. (11) when $\mathbf{A} = 1, \mathbf{B} = 1, \mathbf{S} = 0$ and $\mathbf{U} = \phi$. The computational domain

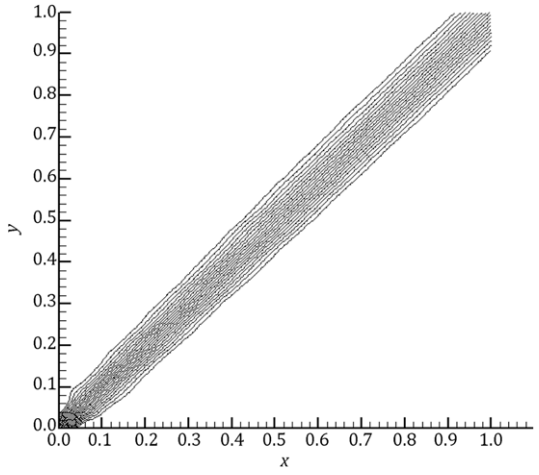


Figure 4: Solution of linear advection with time marching algorithm using irregular nodes.

is discretized using 441 evenly distributed field nodes as illustrated in Figure 1, and 10201 regularly distributed sampling points. $d_w$ is pre-processed to maintain 7 nodes supporting each sampling point, and $P = \begin{bmatrix} 1 & x & y \end{bmatrix}$ is considered. The problem is solved using a steady state algorithm and a time marching one with $\Delta t = 0.1$ s. The obtained results are illustrated in Figure 2.

To assess the accuracy and consistency of the method for irregular nodes, the above mentioned problem is solved using 448 irregular nodes as illustrated in Figure 3. The results of the transient solution with $\Delta t = 0.1$ s, using irregular nodes, are illustrated in Figure 4.

### 4.2. Advection in a spatially varying velocity field

The governing partial differential equation of advection in a spatially varying velocity field is:

$$y\phi_{,x} - x\phi_{,y} = 0 \quad \text{in } -1 \le x \le 1,\ 0 \le y \le 1. \tag{40}$$

The boundary condition is:

$$\phi(-1, y) = 0 \quad \text{on } 0 \le y \le 1, \tag{41}$$
$$\phi(x, 1) = 0 \quad \text{on } 0.5 \le x \le 1, \tag{42}$$
$$\phi(x, 0) = 0 \quad \text{on } -0.35 \le x \le 0, \tag{43}$$
$$\phi(x, 0) = 0 \quad \text{on } -1 \le x \le -0.65, \tag{44}$$
$$\phi(x, 0) = 1 \quad \text{on } -0.65 \le x \le -0.35. \tag{45}$$

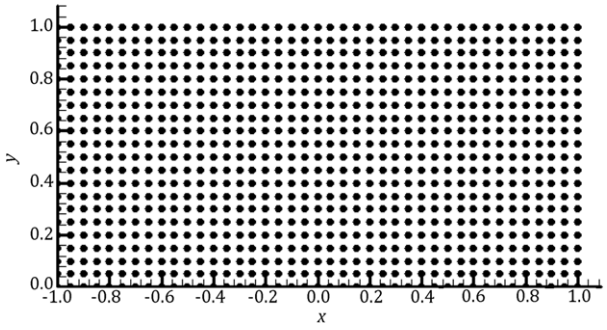


Figure 5: Evenly distributed 882 field nodes.

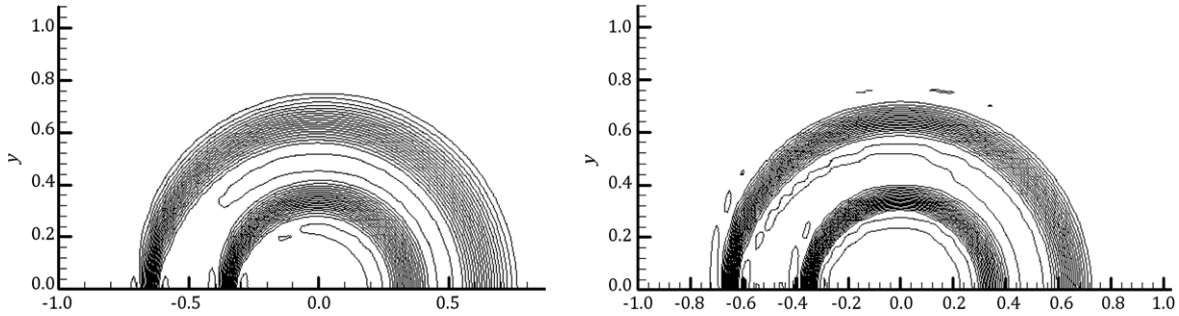


Figure 6: Solution of advection in a spatially varying velocity field.

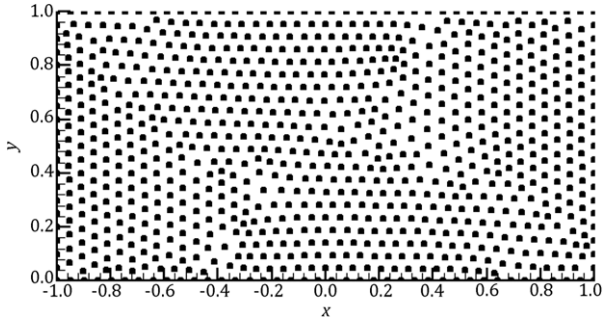


Figure 7: The irregular nodes of advection in a spatially varying velocity field.

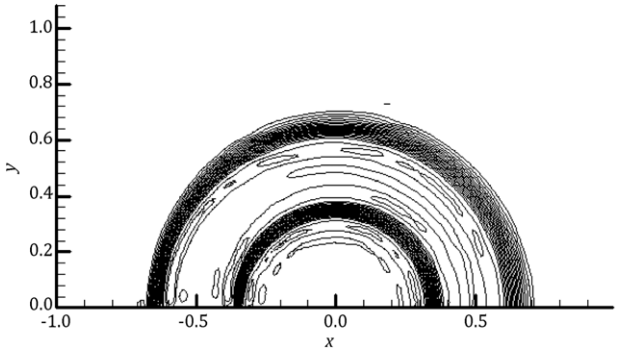


Figure 8: Solution of advection in a spatially varying velocity field with time marching algorithm using irregular nodes.

The exact solution to this problem is known to be:

$$\begin{cases} \phi(x,y) = 1 & 0.35 \le (x^2+y^2)^{0.5} \le 0.65 \\ \phi(x,y) = 0 & \text{otherwise.} \end{cases} \tag{46}$$

The above mentioned partial differential equation is identical to Eq. (11), when $\mathbf{A} = y, \mathbf{B} = -x, \mathbf{S} = 0$ and $\mathbf{U} = \phi$. The computational domain is discretized, using 882 evenly distributed field nodes as illustrated in Figure 5, and 20 301 regularly distributed sampling points. $d_w$ is pre-processed to maintain 7 nodes supporting each sampling point, and $P = \begin{bmatrix} 1 & x & y \end{bmatrix}$ is considered. The problem is solved using a steady state algorithm and a transient one with $\Delta t = 0.1$ s. The obtained results are illustrated in Figure 6.

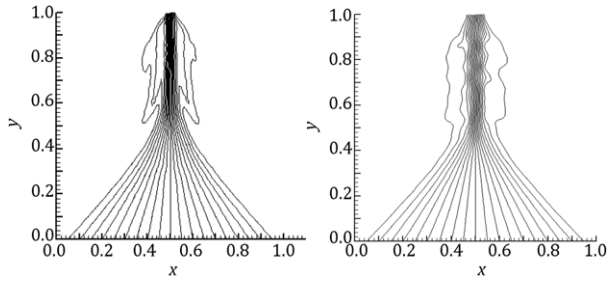


Figure 9: Solution of Burgers equation with transient algorithm.

The same problem is solved using 860 irregular nodes, as illustrated in Figure 7. The time step interval is $\Delta t = 0.1$ s. Other parameters are the same in cases of regular nodes. The numerical solution is presented in Figure 8.

### 4.3. Burgers equation

The steady Burgers equation is considered as an example of a nonlinear equation in the form:

$$\phi\phi_{,x} + \phi_{,y} = 0 \quad \text{in } 0 \le x \le 1,\ 0 \le y \le 1. \tag{47}$$

The boundary condition is:

$$\phi(0,y) = 1 \quad \text{on } 0 \le y \le 1, \tag{48}$$

$$\phi(1,y) = -1 \quad \text{on } 0 \le y \le 1, \tag{49}$$

$$\phi(x,0) = 1 - 2x \quad \text{on } 0 \le x \le 1. \tag{50}$$

The exact solution to this problem consists of two constant states separated by a discontinuity and an expansion fan.

Eq. (47) can be stated in the form of Eq. (11), with $\mathbf{A} = \phi$, $\mathbf{B} = 1$, $\mathbf{S} = 0$ and $\mathbf{U} = \phi$. The computational domain is discretized using 441 evenly distributed field nodes, as illustrated in Figure 1, and 10 201 regularly distributed sampling points. $d_w$ is pre-processed to maintain 7 nodes supporting each sampling point, and $P = \begin{bmatrix} 1 & x & y \end{bmatrix}$ is considered. The problem is solved using a transient solution, with $\Delta t = 0.1$ s. The obtained results are illustrated in Figure 9(a); 448 irregular nodes, as illustrated in Figure 3, are also used to solve this problem, to assess the ability of the method for irregular nodes. The results obtained by these nodes, with $\Delta t = 0.1$ s, are illustrated in Figure 9(b).

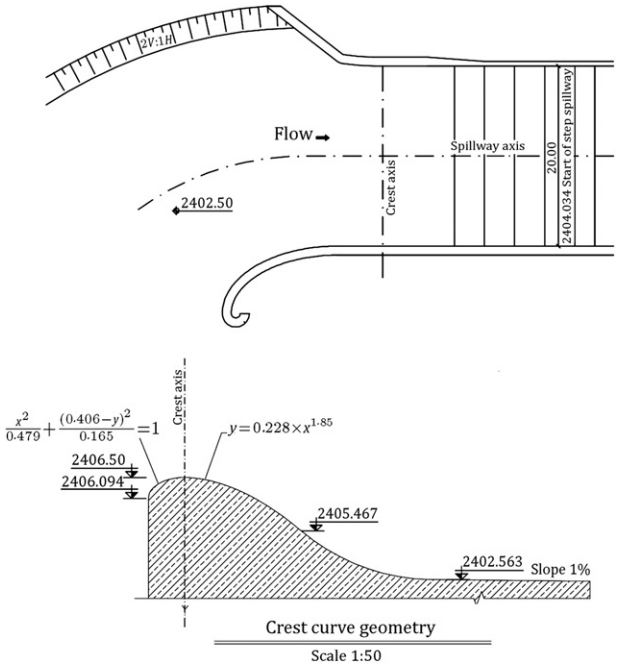


Figure 10: Plan and section of spillway of Siah Bishe Dam.

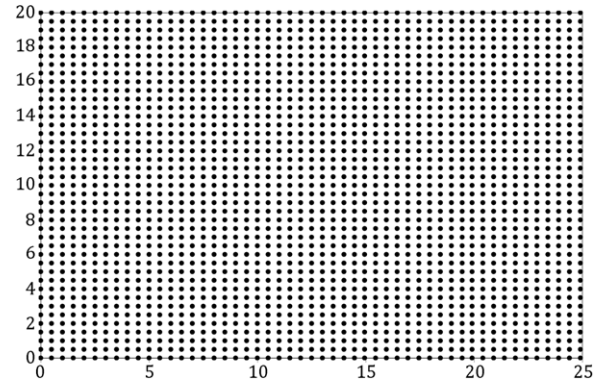

Figure 11: Spatial discretization of 2D regular computational domain.

## 5. Flow over an ogee spillway

In this numerical example, shallow water equations are used to model flow over the spillway of the Siah Bishe Dam. Observed data obtained by a physical model are used to assess the accuracy of results. General specifications of the spillway are illustrated in Figure 10, and observed data are presented in Table 1 [12].

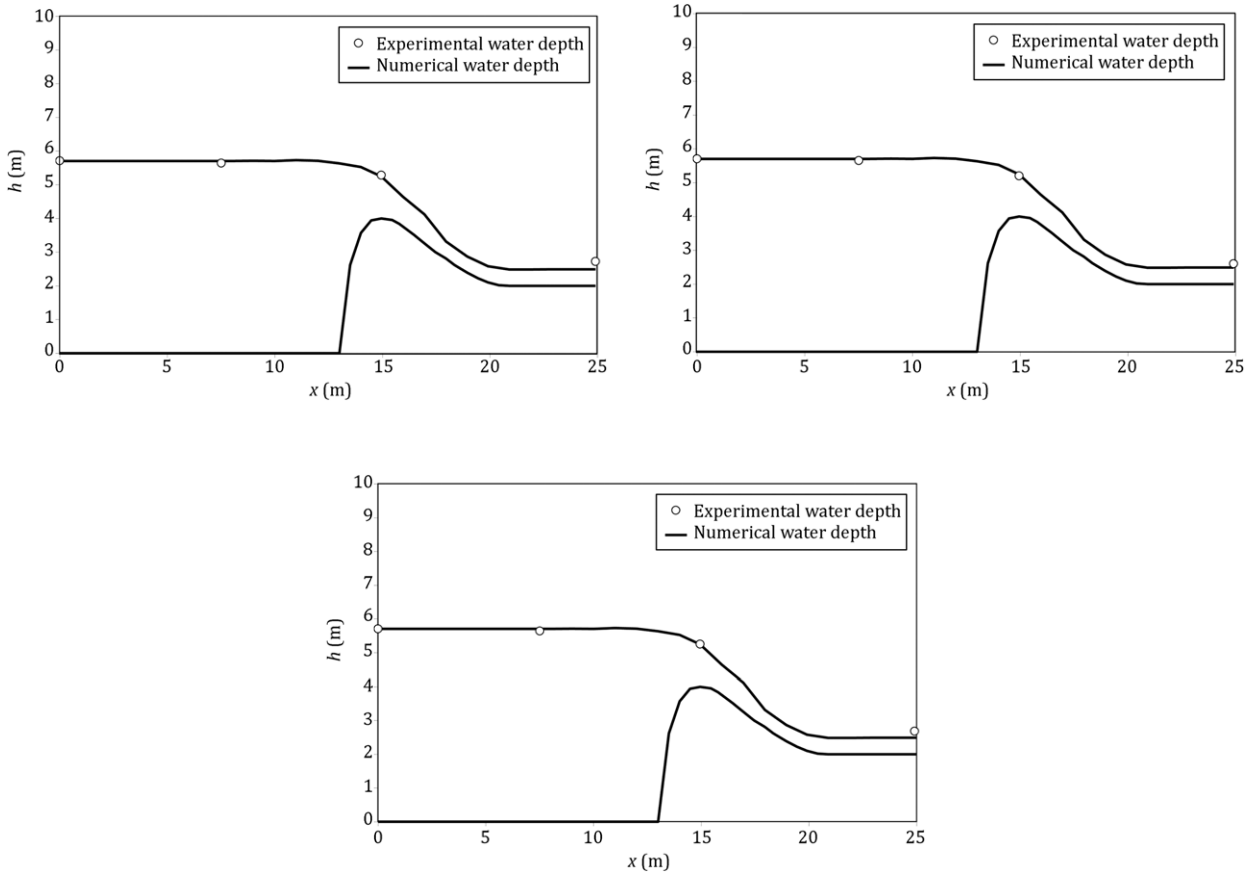


Figure 12: Water depth of 2D regular model.

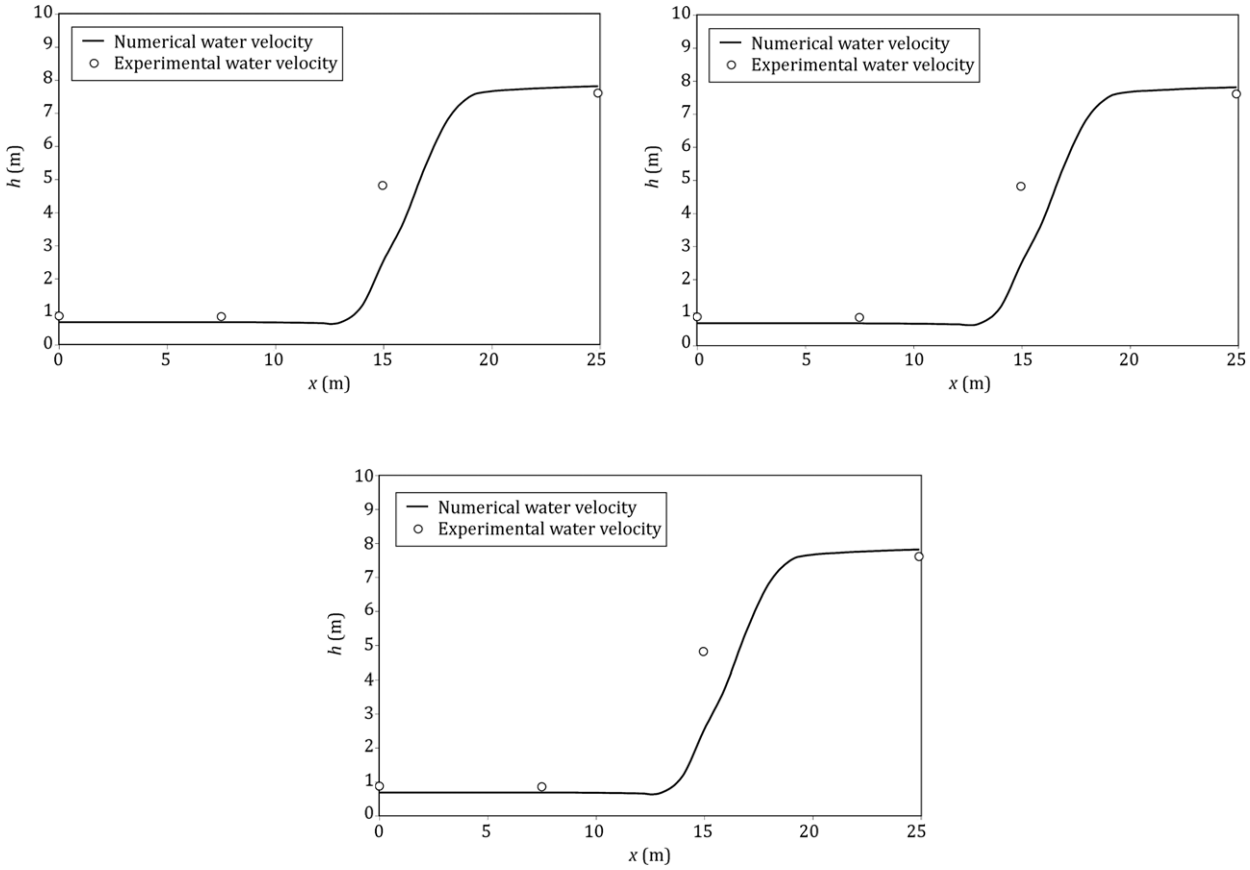


Figure 13: Flow velocity of 2D regular model.

Table 1: Water depth and velocity measurement along the approach channel.

| Siah Bishe hydraulic model sc: 1/15 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Discharge (cm s) | Sections | Specification | | Measurements | Depth from water surface | Location | | |
| | | EL | Distance | | | Right | Center | Left |
| 92.0 | A | 2402.50 | −22.50 | Water depth (m) | | 5.71 | 5.71 | 5.71 |
| | | | | Velocity (m/s) | 0.2 | 0.83 | 0.55 | 0.48 |
| | | | | | 0.8 | 0.62 | 0.39 | 0.36 |
| 92.0 | B | 2402.50 | −15.00 | Water depth (m) | | 5.71 | 5.70 | 5.70 |
| | | | | Velocity (m/s) | 0.2 | 0.86 | 0.63 | 0.45 |
| | | | | | 0.8 | 0.87 | 0.63 | 0.33 |
| 92.0 | C | 2402.50 | −7.50 | Water depth (m) | | 5.64 | 5.65 | 5.64 |
| | | | | Velocity (m/s) | 0.2 | 0.96 | 0.81 | 0.78 |
| | | | | | 0.8 | 0.71 | 0.59 | 0.63 |
| 92.0 | D | 2406.50 | 0.00 | Water depth (m) | | 1.28 | 1.20 | 1.25 |
| | | | | Velocity at 0.6 depth (m/s) | | 4.82 | 4.85 | 4.82 |
| 92.0 | E | 2404.49 | 9.09 | Water depth (m) | | 0.72 | 0.6 | 0.68 |
| | | | | Velocity at 0.6 depth (m/s) | | 7.61 | 7.77 | 7.58 |

This problem is solved in two different cases. First, this problem is solved using a uniform 2D model and, second, a real 2D model is used, as follows.

A model is considered with a width of 20 m and solved in the 2D case. The [0 25 m] × [0 20] computational domain is discretized, using 546 evenly distributed field nodes and 2091 sampling points, as shown in Figure 11. $P = \begin{bmatrix} 1 & x & y \end{bmatrix}$, $\Delta t =$ 0.4 s and $d_w$ is pre-processed to maintain 3 nodes supporting each sampling point. Water depth at the right hand side, center line and left hand side are illustrated in Figure 12. The results obtained for the flow velocity in $y = 0.0$, $y = 10$ and $y = 20$ are presented in Figure 13. As expected, results obtained by the 2D regular model are identical and all three parts in Figures 12 and 13 are the same throughout the symmetry.

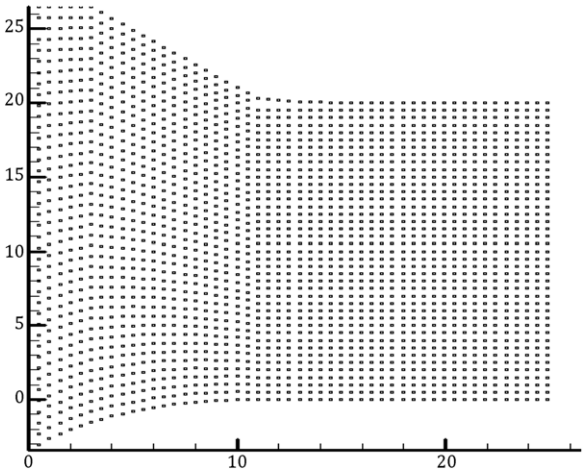


Figure 14: Spatial discretization of 2D computational domain.

At last, for the above mentioned problem, a 2D computational domain is considered and discretized using 546 irregular field nodes, 2091 sampling points, as illustrated in Figure 14, and 141 boundary nodes. Water depth at the right hand side, center line and left hand side are illustrated in Figure 15. Flow velocity at the right hand side, center line and left hand side of the spillway is presented in Figure 16. Streamlines obtained by the DLSM method are illustrated in Figure 17.

## 6. Conclusion

A truly meshless method, the Discrete Least Squares Meshless (DLSM) method, is applied to shallow water flow in this paper. The problem domain is discretized using some nodes, then, using the MLS shape function with a least squares technique, a symmetric stiffness matrix is constructed. In this research, sampling points are defined to minimize the sum of the squared residual functional, and field nodes are defined to construct MLS shape functions. Considering the number of sampling points to be more than the number of field nodes will improve the results and guarantee stability. Results of numerical examples of shallow water problems show good quality for irregular node positioning. This characteristic makes the method really attractive for modeling shallow water flow, whose topography is usually very irregular. In addition, it is notable that this method does not need any background mesh for integration, and increasing the number of sampling points will not increase the dimension of the stiffness matrix.

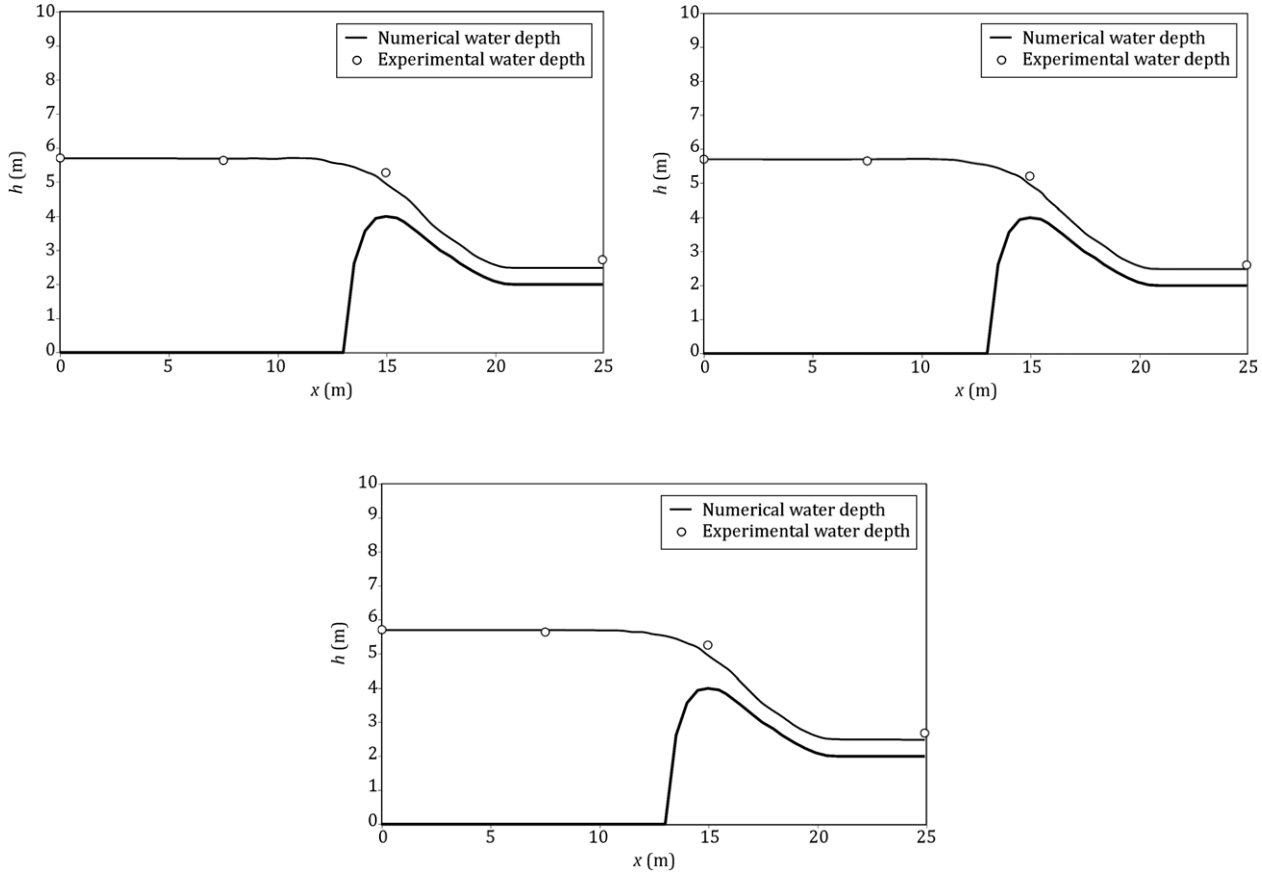


Figure 15: Water depth of 2D irregular model.

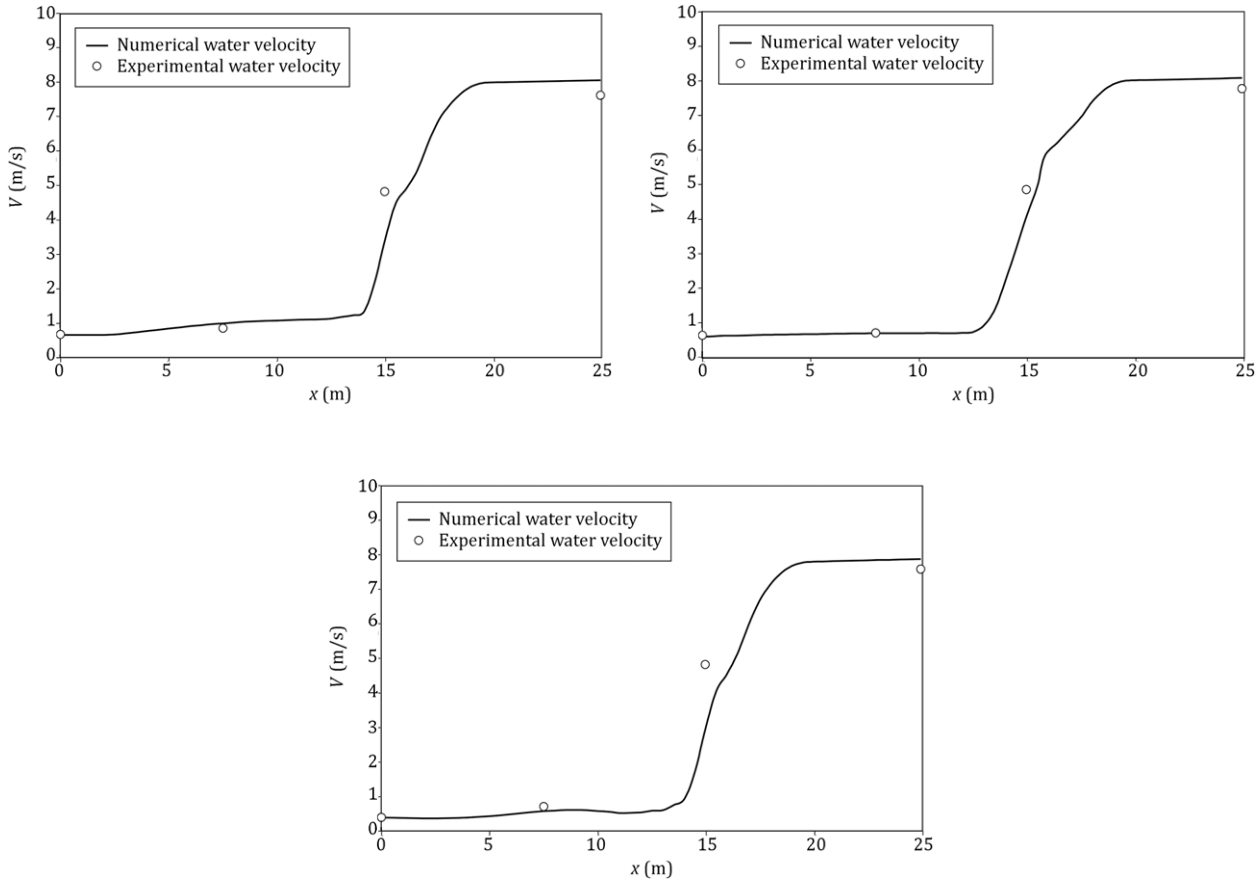


Figure 16: Flow velocity of 2D irregular model.

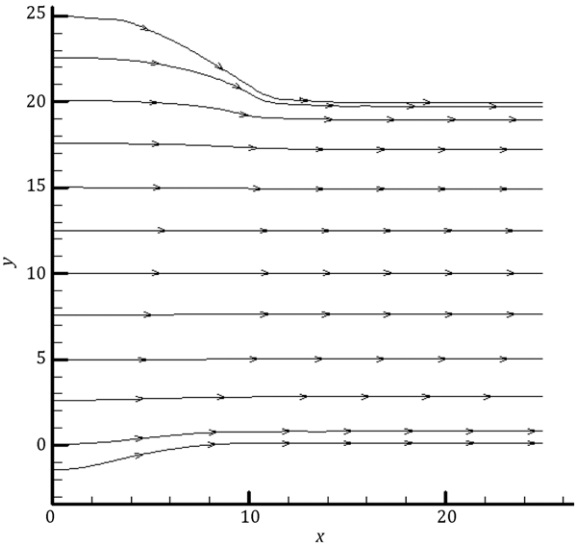


Figure 17: Streamlines obtained by DLSM method.

## References

[1] Cooley, R.L. and Moin, S.A. "Finite element solution of Saint–Venant equations", *Journal of Hydraulic Engineering—ASCE*, 102, pp. 759–775 (1976).
[2] Jimene, O.F. and Chaudhry, M.H. "Computation of supercritical free surface flows", *Journal of Hydraulic Engineering*, 114, pp. 377–395 (1988).
[3] Tingsanchali, T. and Maheswaran, S. "2D depth averaged flow computation near groyne", *Journal of Hydraulic Engineering*, 116, pp. 218–229 (1990).
[4] Lai, C.J. and Yen, C.W. "Turbulent free surface flow simulation using a multilayer model", *International Journal for Numerical Methods in Fluids*, 16, pp. 1007–1025 (1993).
[5] Unami, K., Kawachi, T., Munir, B.M. and Itagaki, H. "Two dimensional numerical model of spillway flow", *Journal of Hydraulic Engineering*, 5, pp. 369–375 (1999).
[6] Aizinger, V. and Dawson, C. "A discontinuous Galerkin method for two dimensional flow and transport in shallow water", *Advances in Water Resources*, 25, pp. 67–84 (2002).
[7] Zhou, X., Hon, Y.C. and Cheung, K.F. "A grid-free, nonlinear shallow-water model with moving boundary", *Journal of Engineering Analysis with Boundary Elements*, 28, pp. 967–973 (2004).
[8] Arzani, H. "A meshless method for the solution of shallow water equations", Doctoral Dissertation, School of Civil Engineering, Iran University of Science and Technology, Tehran (2006).
[9] Darbani, M., Ouahsine, A., Villon, P., Naceur, H. and Smaoui, H. "Meshless method for shallow water equations with free surface flow", *Applied Mathematics and Computation*, 217, pp. 5113–5124 (2011).
[10] Rodriguez-Paz, M. and Bonet, J. "A corrected smooth particle hydrodynamics formulation of the shallow-water equations", *Computers & Structures*, 83, pp. 1396–1410 (2005).
[11] Gingold, R.A. and Monaghan, J.J. "Smoothed particle hydrodynamics: Theory and applications to non-spherical stars", *Monthly Notices of the Royal Astronomical Society*, 181, pp. 375–389 (1977).
[12] Belytschko, T., Lu, Y. and Gu, L. "Element free Galerkin methods", *International Journal for Numerical Methods in Engineering*, 37, pp. 229–256 (1994).
[13] Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. and Krysl, P. "Meshless methods: an overview and recent developments", *Computer Methods in Applied Mechanics and Engineering*, 139, pp. 3–47 (1996).
[14] Liu, W.K., Jun, S. and Zhang, Y.F. "Reproducing kernel particle methods", *International Journal for Numerical Methods in Fluids*, 20, pp. 1081–1106 (1995).
[15] Onate, E., Idelsohn, S., Zienkiewicz, O.C. and Taylor, R.L. "A finite point method in computational mechanics. Applications to convective transport and fluid flow", *International Journal for Numerical Methods in Engineering*, 39(22), pp. 3839–3866 (1996).
[16] Liszka, T.J., Duarte, C.A.M. and Tworzydlo, W.W. "hp-meshless cloud method", *Computer Methods in Applied Mechanics and Engineering*, 139, pp. 263–288 (1996).

[17] Atluri, S.N. and Zhu, T.L. "The meshless local Petrov–Galerkin (MLPG) approach for solving problems in elasto-statics", *Computational Mechanics*, 25, pp. 169–179 (2000).
[18] Atluri, S.N. and Zhu, T.L. "New concepts in meshless methods", *International Journal for Numerical Methods in Engineering*, 47(1–3), pp. 537–556 (2000).
[19] Atluri, S.N. and Zhu, T. "A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics", *Computational Mechanics*, 22, pp. 117–127 (1998).
[20] Long, S. and Atluri, S.N. "A meshless local Petrov–Galerkin method for solving the bending problem of a thin plate", *CMES Computer Modeling in Engineering & Sciences*, 1, pp. 53–63 (2002).
[21] Atluri, S.N., Kim, H.G. and Cho, J.Y. "A critical assessment of the truly meshless local Petrov–Galerkin (MLPG), and local boundary integral equation (LBIE) methods", *Computational Mechanics*, 24, pp. 348–372 (1999).
[22] Atluri, S.N., Sladek, J., Sladek, V. and Zhu, T. "The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity", *Computational Mechanics*, 25, pp. 180–198 (2000).
[23] Zhu, T., Zhang, J.D. and Atluri, S.N. "A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach", *Computational Mechanics*, 21, pp. 223–235 (1998).
[24] Aluru, N.R. and Gang, L. "Finite cloud method: a true meshless technique based on a fixed reproducing kernel approximation", *International Journal for Numerical Methods in Engineering*, 50, pp. 2373–2410 (2001).
[25] Pan, X.F., Zhang, X. and Lu, M.W. "Meshless Galerkin least-square method", *Computational Mechanics*, 35, pp. 182–189 (2005).
[26] Zhang, X., Liu, X.H., Song, K.Z. and Lu, M.W. "Least-squares collocation meshless method", *International Journal for Numerical Methods in Engineering*, 51, pp. 1089–1100 (2001).
[27] Arzani, H. and Afshar, M.H. "Solving Poisson's equations by the discrete least square meshless method", *Proceeding of 28th Boundary Elements and other Mesh Reduction Methods (BEM/MRM28)*, Skiathos, Greece (2006).
[28] Arzani, H. and Afshar, M.H. Solution of spillways flow by discrete least square meshless methods", *Proceeding of Second ECCOMAS Thematic Conference on Meshless Methods*, Porto, Portugal (2007).
[29] Firoozjaee, A.R. and Afshar, M.H. "Discrete least squares meshless method (DLSM) with sampling points for the solution of elliptic partial differential equations", *Journal of Engineering Analysis with Boundary Elements*, 33(1), pp. 83–92 (2009).
[30] Afshar, M.H. and Lashckarbolok, M. "Collocated discrete least-squares (CDLS) meshless method: error estimate and adaptive refinement", *International Journal for Numerical Methods in Fluids*, 56(10), pp. 1909–1928 (2008).
[31] Shepard, D. Two-dimensional interpolation function for irregularly-spaced data", in: *23rd National Conference ACM*, pp. 517–524 (1968).
[32] Zienkiewicz, O.C. and Taylor, R.L., *The Finite Element Methods: Fluid dynamics*, 5th ed., vol. 3, Butterworth–Heinemann, Oxford (2000).
[33] Du, C.J. "An element-free Galerkin method for simulation of stationary two-dimensional shallow water flows in rivers", *Computer Methods in Applied Mechanics and Engineering*, 182, pp. 89–107 (2000).

**A. Rahmani Firoozjaee** received his B.S. degree from Tabriz University, Iran, in 1995, his M.S. degree from Kerman University, Iran, in 1998, and his Ph.D. degree from Iran University of Science and Technology (IUST), Tehran, in 2010. He is currently Assistant Professor in the Department of Civil Engineering at Babol University of Technology, Iran. He has authored more than 20 technical publications, including 8 international journal papers. His research interests include computational fluid dynamics (CFD), computational mechanics and engineering optimization.

**M.H. Afshar** received his B.S. degree from the Engineering Faculty at Tehran University, Iran in 1984 and his M.S. and Ph.D. degrees from the University College of Swansea, UK in 1989 and 1993, respectively, all in Civil Engineering. He is currently Associate Professor of Civil Engineering at the Iran University of Science and Technology.

He worked as part time researcher with Mahab Ghods consulting Engineers from 1996 to 1999, where he managed two research projects on the Pashakola Dam Break and Wave Energy Estimation in the Chahbahar region. Moreover, he has managed eleven research projects on the optimal design of water distribution networks, sewer networks, pipeline systems, flood controllingsystems, reservoir operation, spillway flow and spillway design, leading to the development of OPTNET software for the optimal design of water distribution networks.

He is author or coauthor of over ninety research articles in national and international journals and seventy conference papers. He has translated several books into Persian, including "Finite Element and Approximation", "The Finite Element Method" (Vol. I & II), and "Civil Engineering Systems". He has also supervised about 40 postgraduate students, three of which are Ph.D. candidates. His research interests include Mathematical Modeling and Optimization in Hydraulic Structures and Water Engineering.